

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

Лабораторная работа № 6
по программированию
(Вариант 1537)

Студент
Дубинин Артём Сергеевич
группа Р3115

Преподаватель
Карасева Мария Александровна

Санкт - Петербург 2023 год

Оглавление

Задание.....	3
Диаграмма классов объектной модели:.....	4
Вывод.....	5

Задание

Разделить программу из [лабораторной работы №5](#) на клиентский и серверный модули. Серверный модуль должен осуществлять выполнение команд по управлению коллекцией. Клиентский модуль должен в интерактивном режиме считывать команды, передавать их для выполнения на сервер и выводить результаты выполнения.

Необходимо выполнить следующие требования:

- Операции обработки объектов коллекции должны быть реализованы с помощью Stream API с использованием лямбда-выражений.
- Объекты между клиентом и сервером должны передаваться в сериализованном виде.
- Объекты в коллекции, передаваемой клиенту, должны быть отсортированы по местоположению
- Клиент должен корректно обрабатывать временную недоступность сервера.
- Обмен данными между клиентом и сервером должен осуществляться по протоколу UDP
- Для обмена данными на сервере необходимо использовать **сетевой канал**
- Для обмена данными на клиенте необходимо использовать **датаграммы**
- Сетевые каналы должны использоваться в неблокирующем режиме.

Обязанности серверного приложения:

- Работа с файлом, хранящим коллекцию.
- Управление коллекцией объектов.
- Назначение автоматически генерируемых полей объектов в коллекции.
- Ожидание подключений и запросов от клиента.
- Обработка полученных запросов (команд).
- Сохранение коллекции в файл при завершении работы приложения.
- Сохранение коллекции в файл при исполнении специальной команды, доступной только серверу (клиент такую команду отправить не может).

Серверное приложение должно состоять из следующих модулей (реализованных в виде одного или нескольких классов):

- Модуль приёма подключений.
- Модуль чтения запроса.
- Модуль обработки полученных команд.
- Модуль отправки ответов клиенту.

Сервер должен работать в **однопоточном** режиме.

Обязанности клиентского приложения:

- Чтение команд из консоли.
- Валидация вводимых данных.
- Сериализация введенной команды и её аргументов.
- Отправка полученной команды и её аргументов на сервер.

- Обработка ответа от сервера (вывод результата исполнения команды в консоль).
- Команду **save** из клиентского приложения необходимо убрать.
- Команда **exit** завершает работу клиентского приложения.

Важно! Команды и их аргументы должны представлять из себя объекты классов.

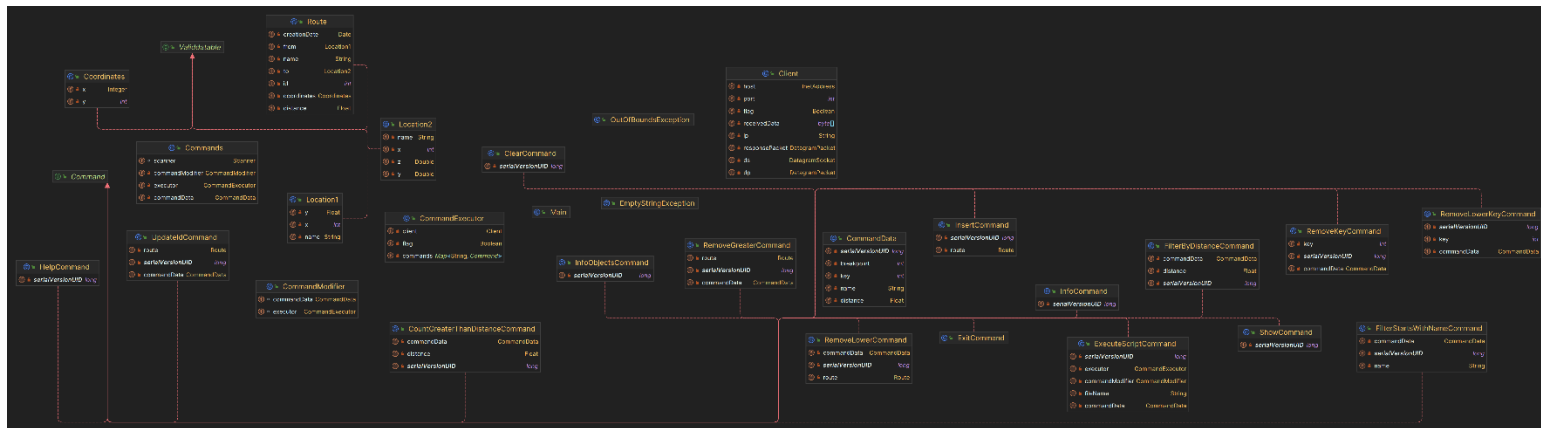
Недопустим обмен "простыми" строками. Так, для команды add или её аналога необходимо сформировать объект, содержащий тип команды и объект, который должен храниться в вашей коллекции.

Дополнительное задание:

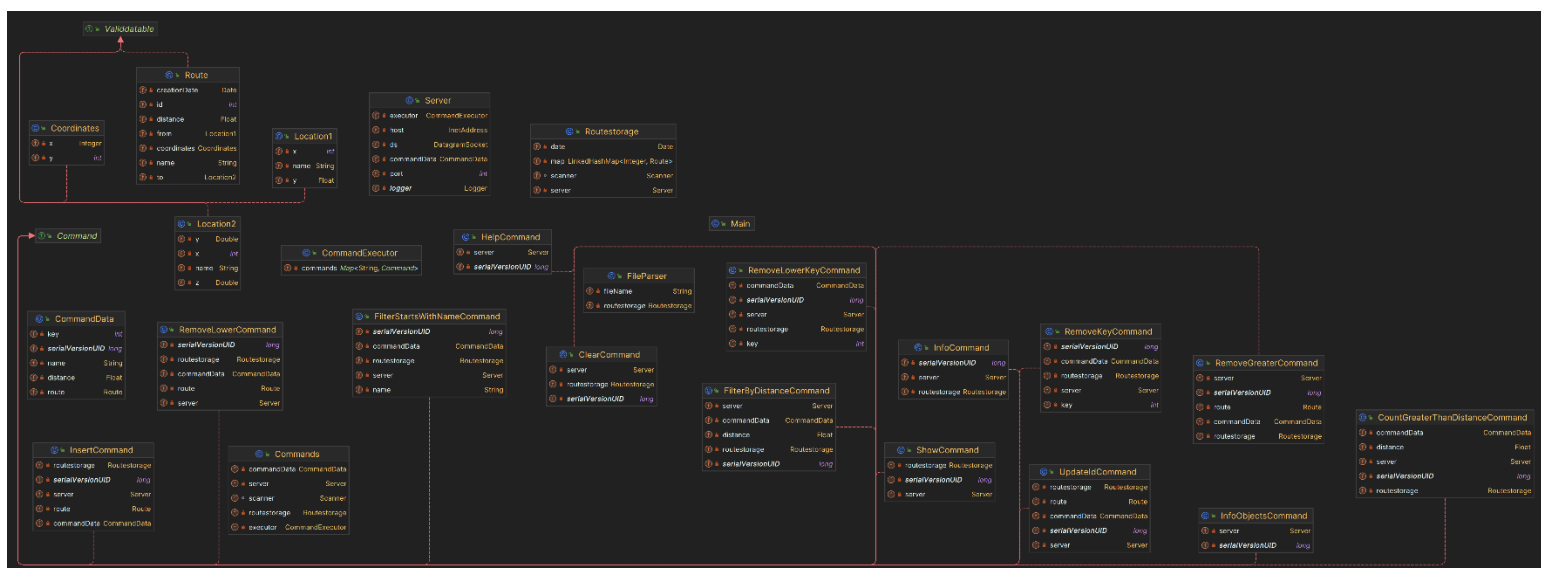
Реализовать логирование различных этапов работы сервера (начало работы, получение нового подключения, получение нового запроса, отправка ответа и т.п.) с помощью **Logback**

Диаграмма классов объектной модели:

Клиент



Сервер



Вывод

Во время разработки клиент-серверного приложения я познакомился с протоколом UDP. Также я проработал структуру сервера и клиента и организовал взаимодействие между ними, углубил знания о логировании в Java